

Creation of Repositories

1 Contents

1	Preface.....	2
2	Prerequisites.....	2
3	Creation of Repository	2
4	Cloning of Repository	3
5	Creation of Data Storage	3
6	Set Roles and Permissions.....	4
6.1	Repository Permissions	4
6.2	Initial Object ACL	5
6.3	Assignment of Roles to Users and Groups	6
6.4	Object Permissions	7
7	Repository Status	7
8	Assignment of Metamodel to Repository	8
8.1	QIS and QLM Metamodels	8
8.2	QIS Metamodel.....	8
8.2.1	Development Mode.....	9
8.2.2	Production Mode.....	9
8.2.3	C# Scripts.....	17
8.3	QLM Metamodel	18
9	Synchronization of Shared Attributes	19
10	Creation of Configurations	20
11	Access Control	21
12	Remote Repositories	24
13	QCL Engine Manager	25
14	Addons.....	25
15	Service Operations	26
16	Object Reservations	27

1 Preface

This document describes the process of creation and configuration of repository.

2 Prerequisites

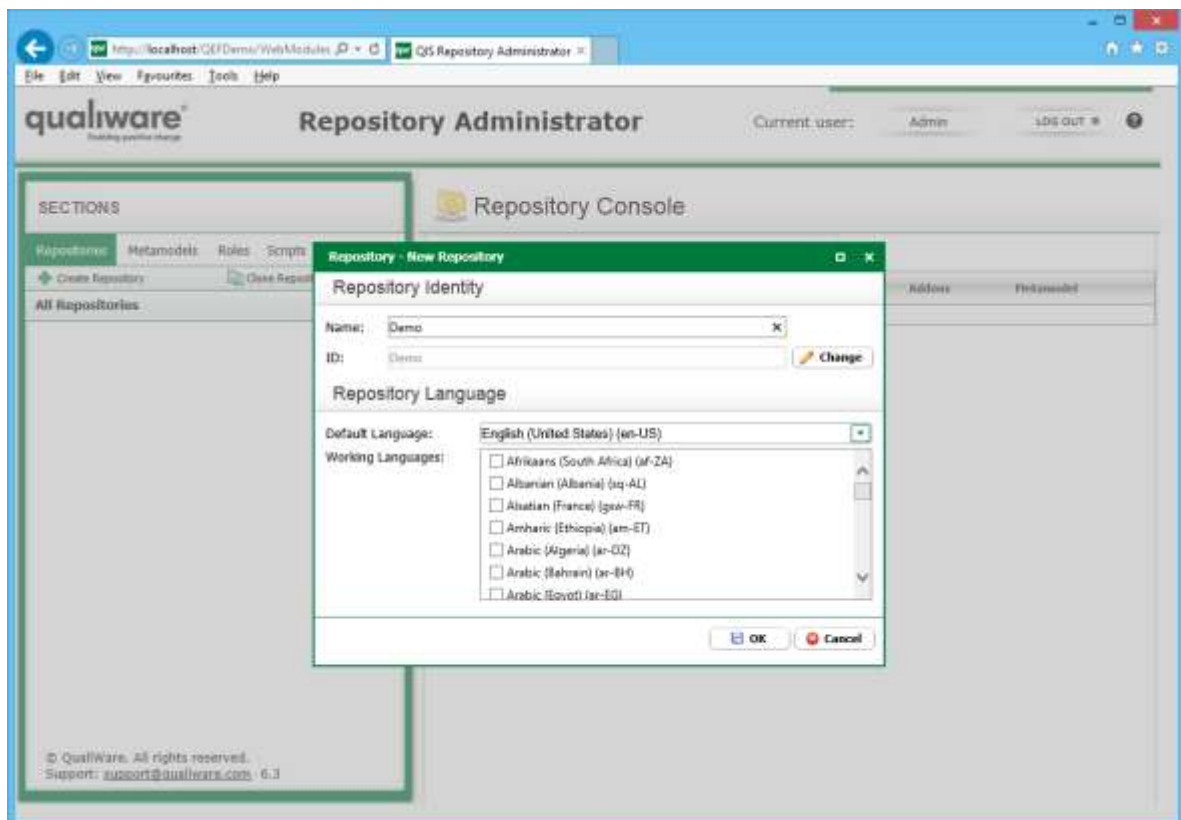
QualiWare Execution Framework (hereinafter QEF) and Admin Console are installed and running.

QualiWare Integration Server (hereinafter QIS) and **Repository Administrator** (hereinafter RA) modules are installed and running.

User is assigned **QIS Administrator** role. Some operations may require user be assigned **QEF Administrator** role.

3 Creation of Repository

1. Go to RA -> Repositories.
2. Click **Create Repository**.
3. Enter name of repository. By default name and ID of repository are the same.
4. Select default and working languages.
5. Click **OK**.

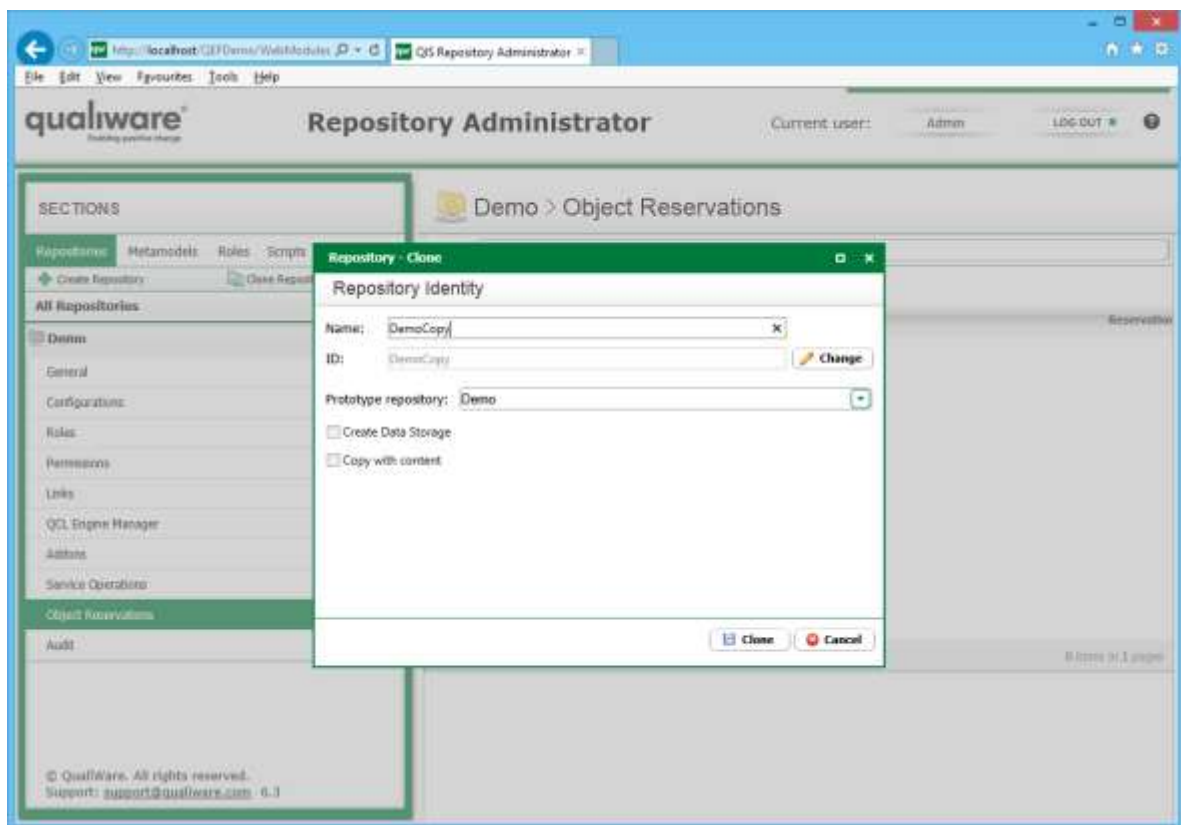


Notes:

- Change of ID of repository after it is created and it contains data, may break remote repository object links if such exist.
- Default language cannot be changed after repository is created. However it is possible to replace it (see [Service Operations](#)).

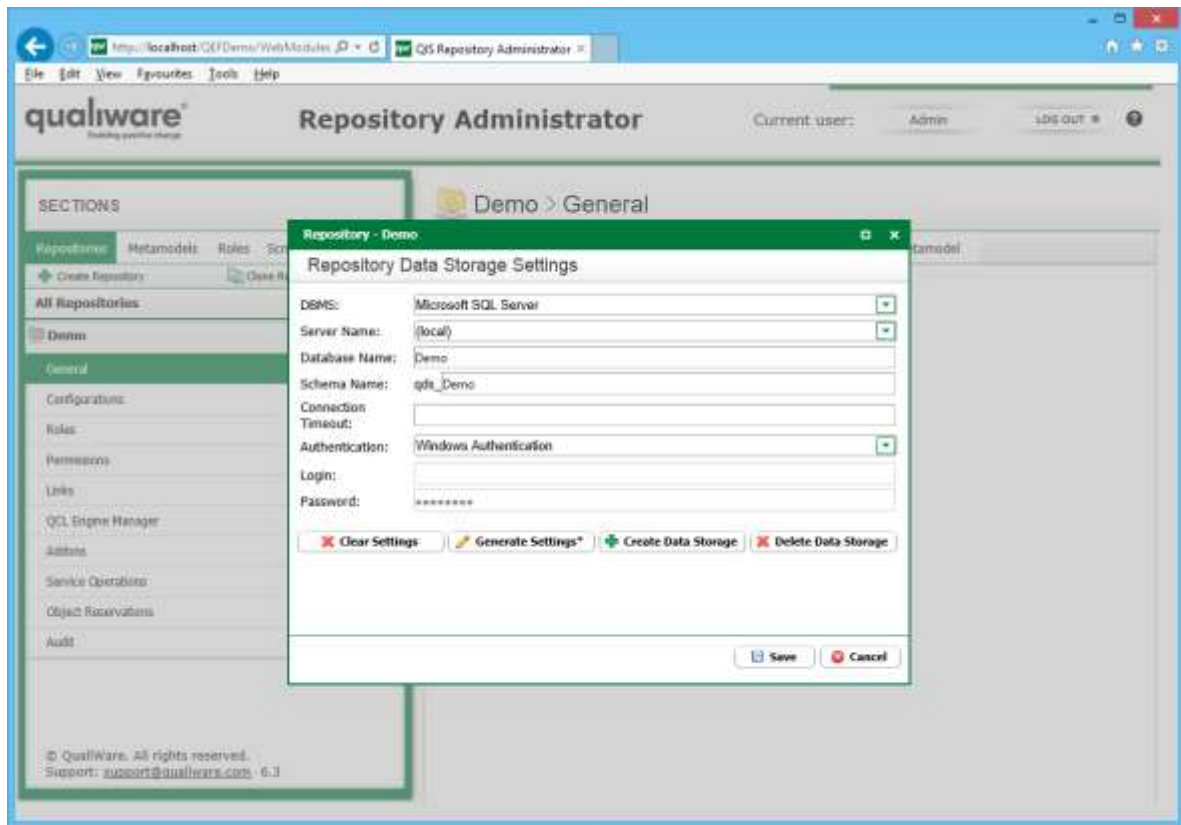
4 Cloning of Repository

1. Go to RA -> Repositories.
2. Click **Clone Repository**.
3. Enter name of repository. By default name and ID of repository are the same.
4. Select if data storage should be created and data copied.
5. Click **Clone**.



5 Creation of Data Storage

1. Go to RA -> Repositories -> [Repository] -> General -> Data Storage Settings.
2. Click **Generate Settings** or enter settings manually.
3. Click **Create Data Storage**.
4. Click **Save**.



Notes:

- More than one repository data storage can be created in a single database. Each repository data storage must use unique schema name.

6 Set Roles and Permissions

Permissions are defined on the following levels:

- Repository permissions.
- Initial object ACL.
- Object permissions.

6.1 Repository Permissions

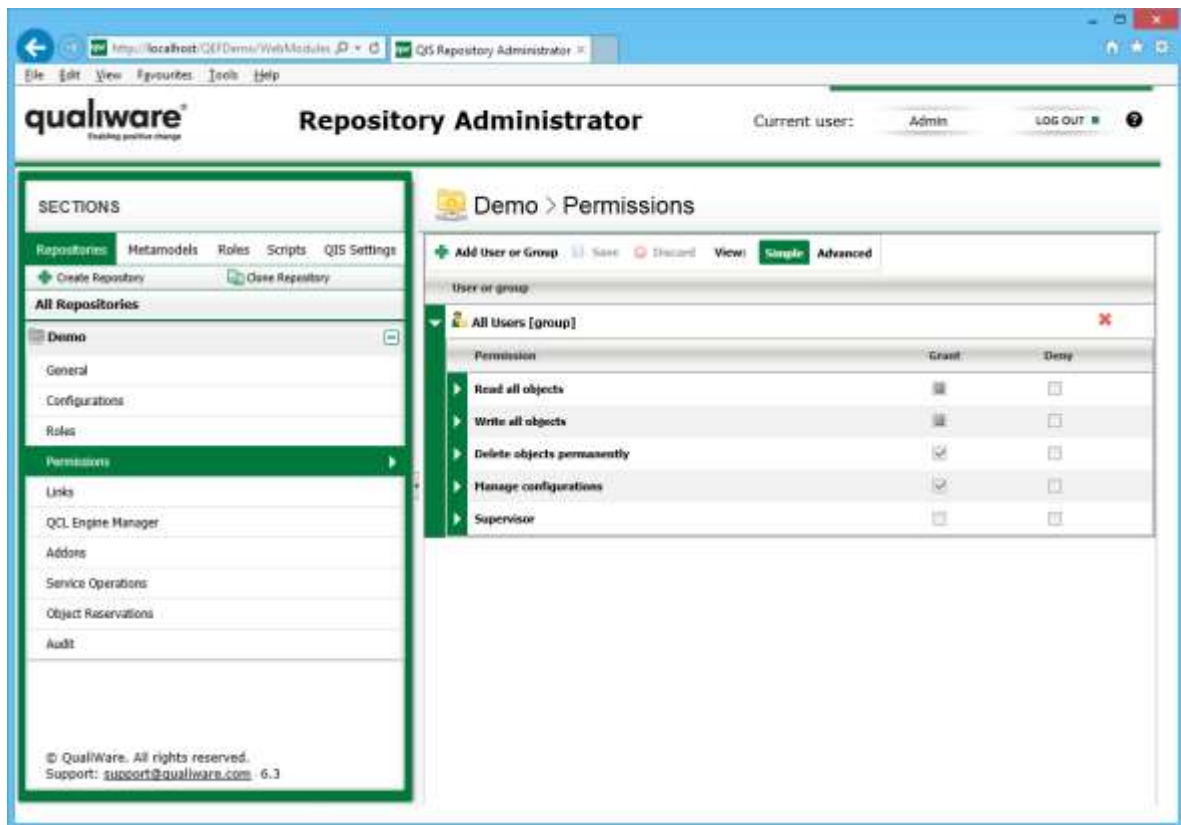
Repository permissions are not assigned to objects directly, but calculated when user connects to repository. Because they are not assigned to objects directly, it is possible to change repository permissions after objects are created and changes will have effect on existing and future objects. To change repository permissions:

1. Go to RA -> Repositories -> {Repository} -> Permissions.

2. Expand group or user node to modify existing assignments or click **Add User of Group** to add new assignment.
3. Modify permissions as required.

Note: Deny permission always wins over grant.

4. Click **Save**.



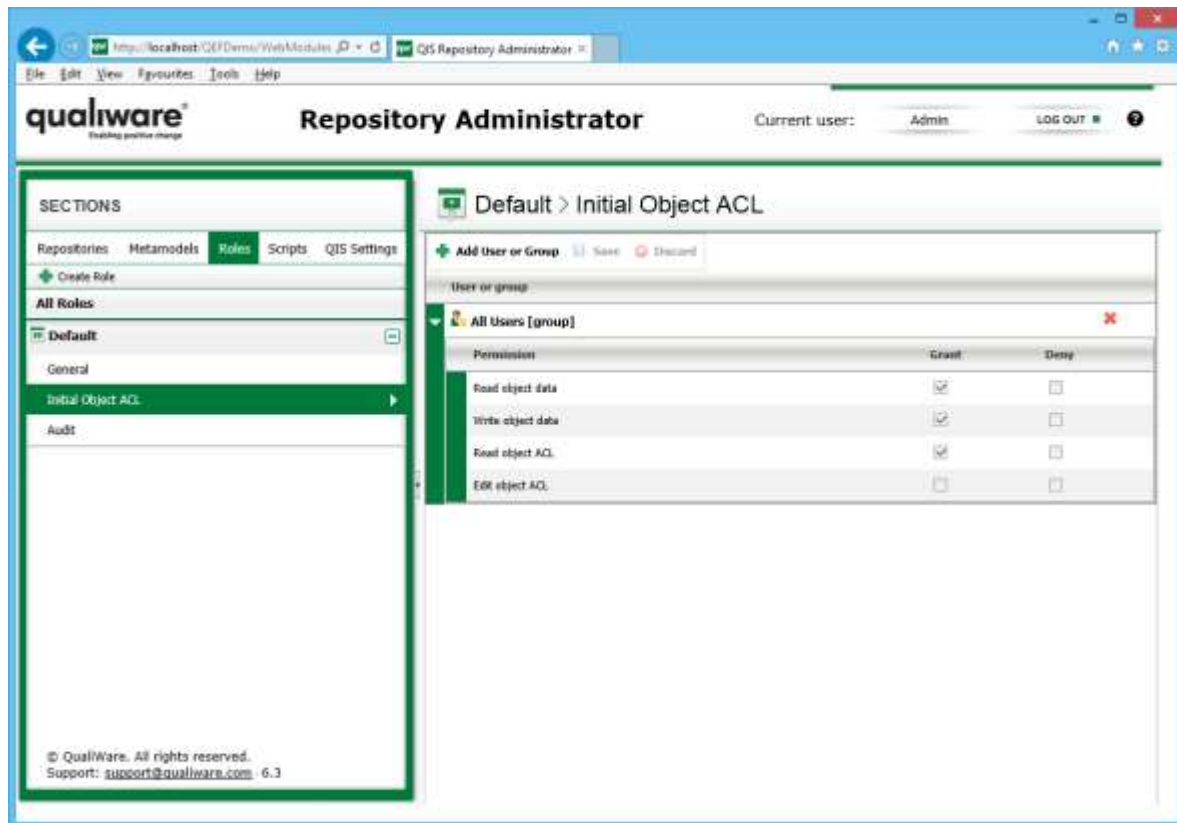
6.2 Initial Object ACL

Initial object ACL are assigned to object at the moment when it is created. These permissions overwrite repository permissions. Because they are assigned to objects directly, any change to initial object ACLs after objects are created will not have effect on them, but only on future objects. Existing objects will need to have ACLs re-assigned accordingly using QIS API or QLM. To change initial object ACLs:

1. Go to RA -> Roles -> {Role} -> Initial Object ACL.
2. Expand group or user node to modify existing assignments or click **Add User of Group** to add new assignment.
3. Modify permissions as required.

Note: Deny permission always wins over grant.

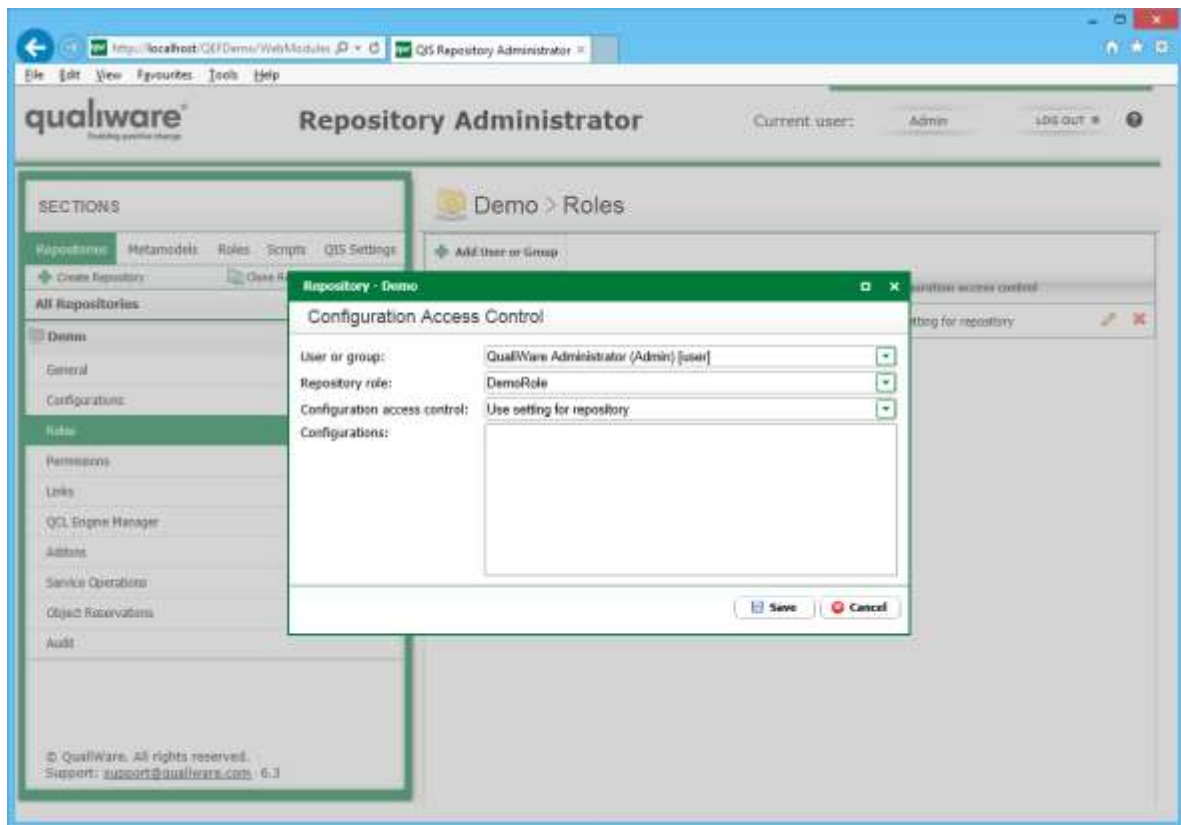
4. Click **Save**.



6.3 Assignment of Roles to Users and Groups

Roles need to be explicitly assigned to user or group for users be able to connect with the specified roles. To assign roles to users and groups, follow the steps:

1. Go to RA -> Repositories -> [Repository] -> Roles.
2. Click **Add User or Group**.
3. Select user or group.
4. Select role.
5. Click **Save**.



6.4 Object Permissions

Object permissions are assigned to objects directly by means of QIS API or QLM.

7 Repository Status

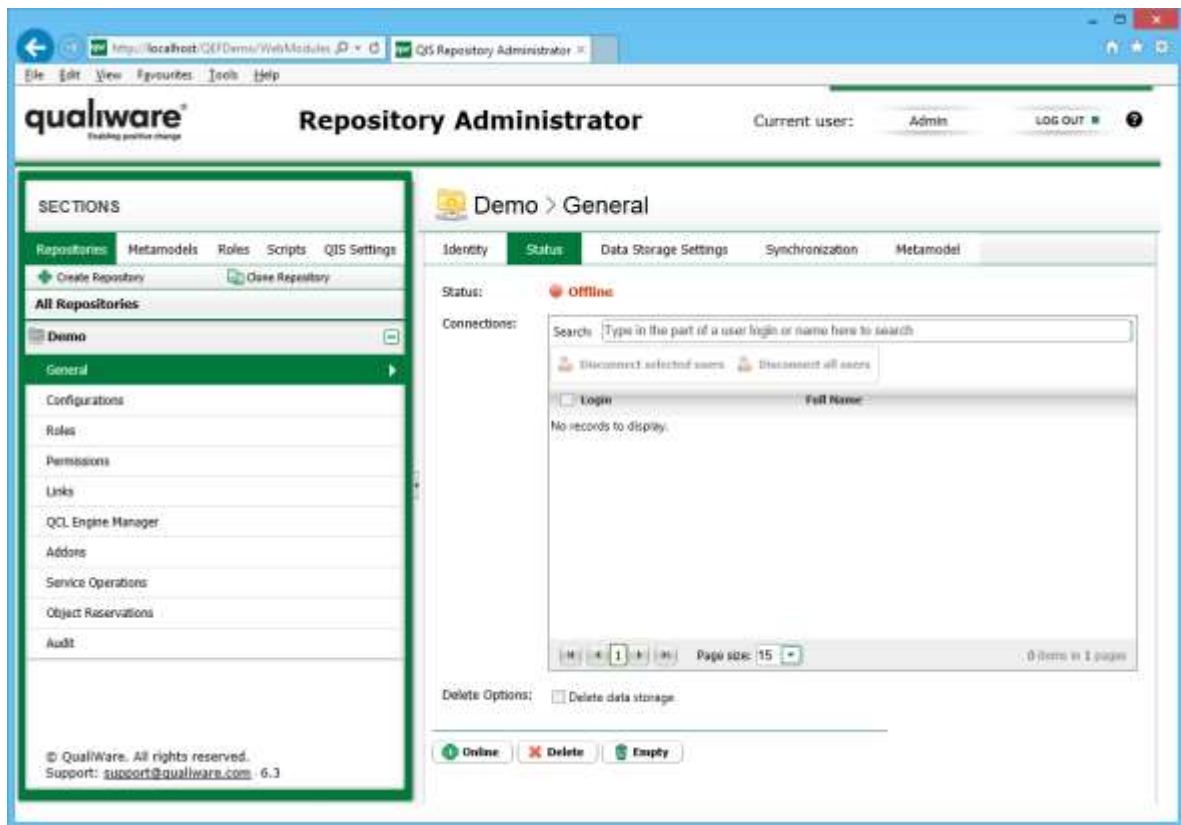
Repository can be in online or offline state.

Online state allows multiple users to connect.

Offline state allows single user to connect. Offline state is designed for maintenance tasks like defining data storage and metamodel, dump loading and others.

To change repository state, follow the steps:

1. Go to RA -> Repositories -> {Repository} -> General -> Status.
2. Click **Offline** or **Online**, depending on the current state.



8 Assignment of Metamodel to Repository

8.1 QIS and QLM Metamodels

QIS and QLM metamodels are defined separately and used by QIS and QLM sole and respectively. Both metamodels are in different formats, having some similarities. Both metamodels are generated by Casemaker at the same time.

Despite the fact that QIS and QLM metamodels are two separate metamodels, it is possible to configure QLM to use its metamodel through QIS. This configuration is not default and should explicitly be turned on. To configure QLM to use its metamodel through QIS, follow the steps:

1. Open `models\openrep.qcl` file.
2. Find **UseQisMetamodel** setting and set its value to one of the following:
 - 0 (default) - Use QLM metamodel.
 - 1 - Use QIS metamodel.

8.2 QIS Metamodel

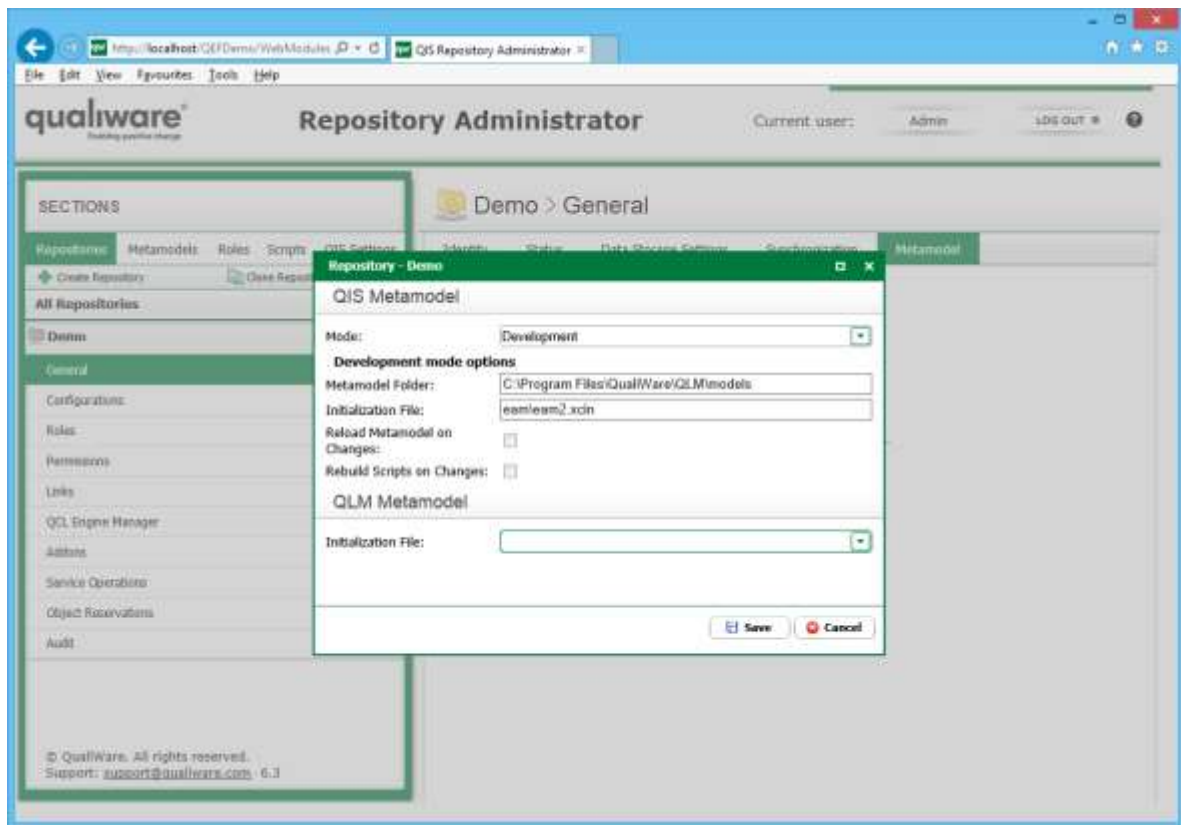
QIS metamodel supports two modes:

- **Development mode.** It takes metamodel from Models folder in file system.
- **Production mode.** It takes metamodel from database, where it must preliminary be deployed.

8.2.1 Development Mode

To configure repository to use metamodel in development mode, follow the steps:

1. Go to RA -> Repositories -> {Repository} -> General -> Metamodel.
2. Click **Edit**.
3. Select **Development** mode.
4. Enter full path to Models folder.
5. Enter relative path to .xcin file under Models folder.
6. Click **Save**.



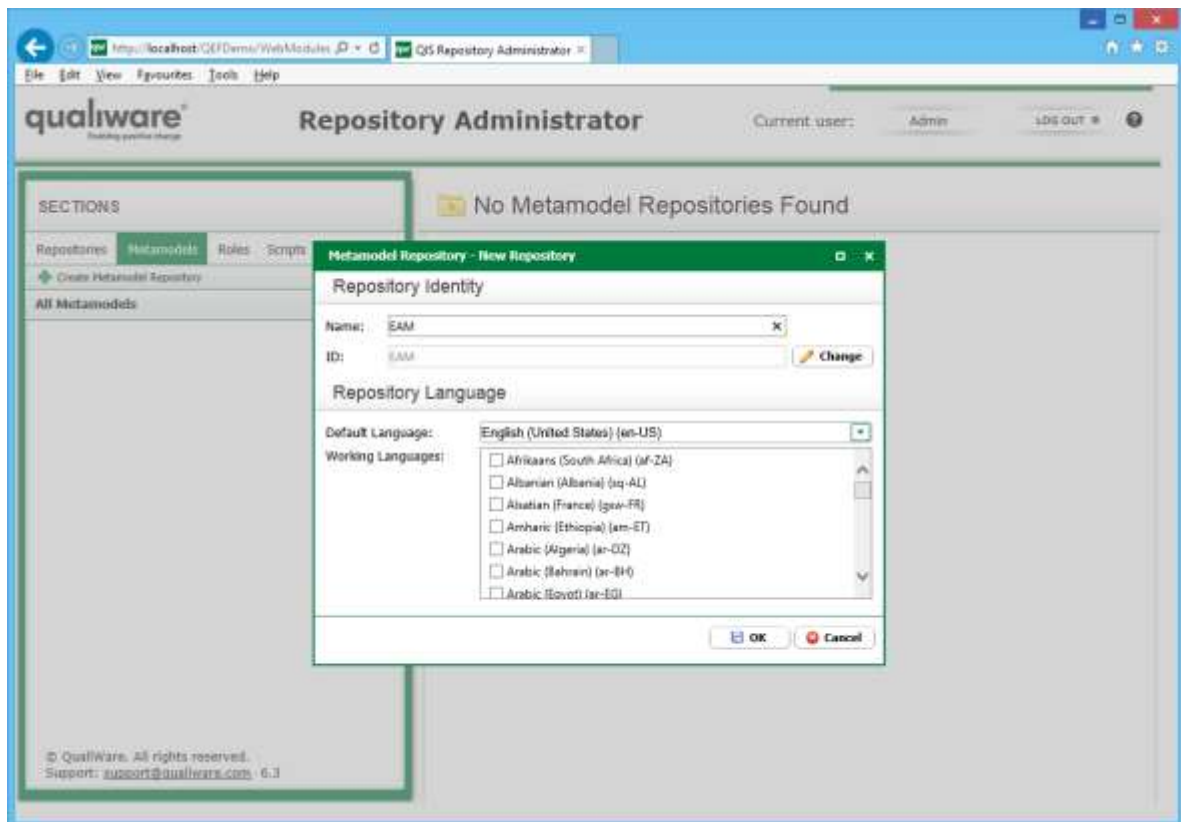
8.2.2 Production Mode

Configuration of repository to use metamodel in production mode includes the following steps:

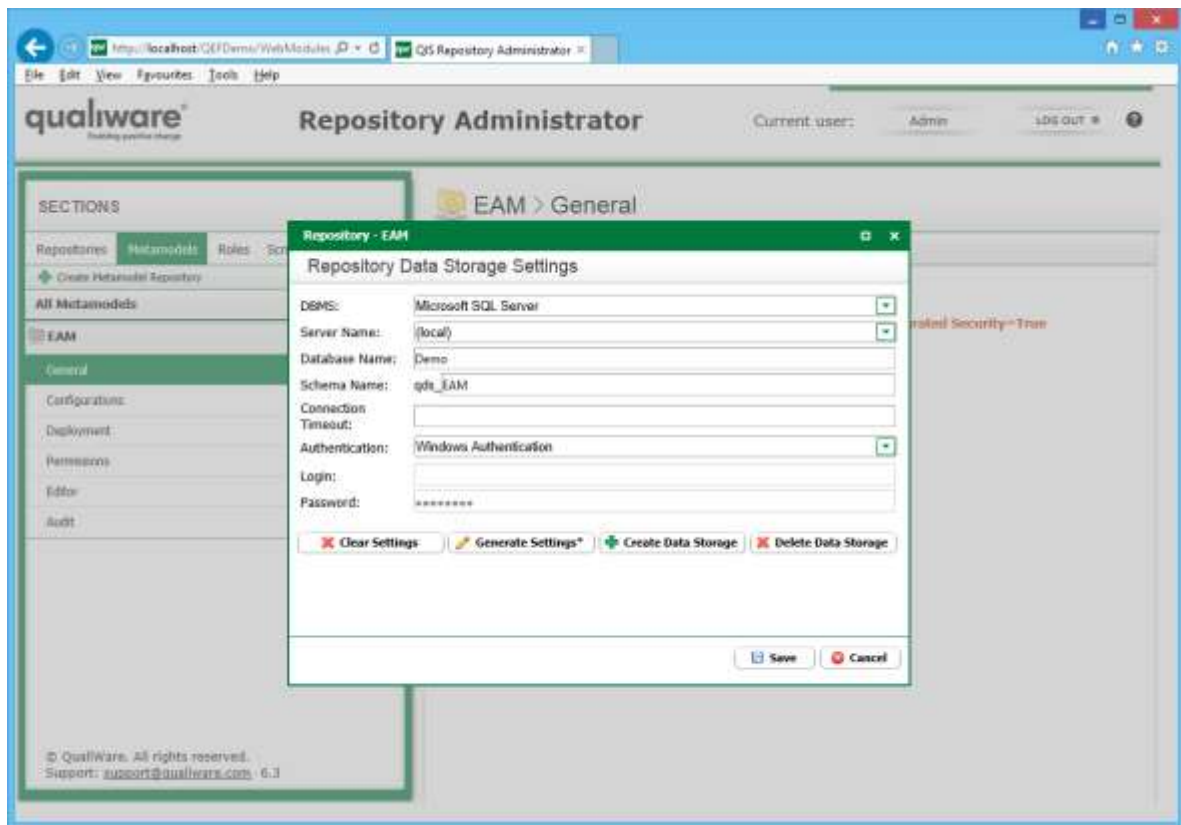
1. Creation of metamodel repository.
2. Deployment of metamodel.
3. Assignment of metamodel.

8.2.2.1 Creation of Metamodel Repository

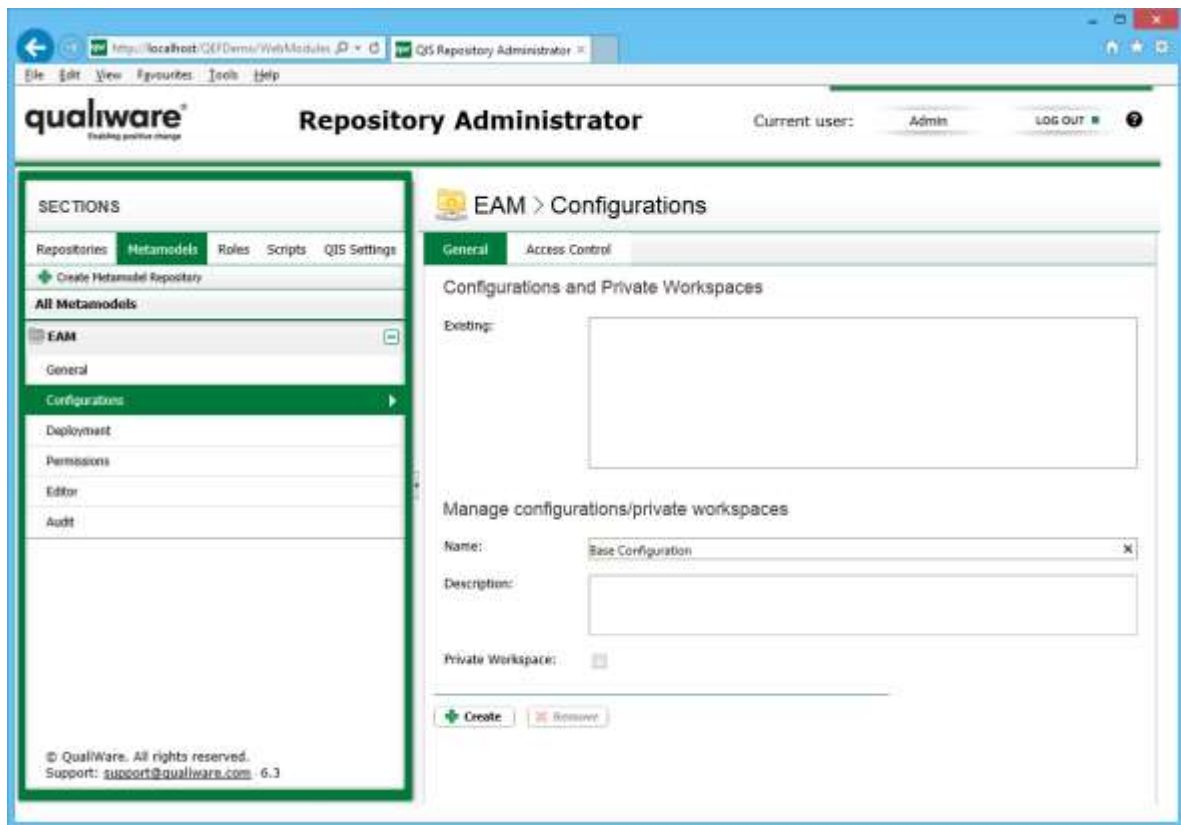
1. Go to RA -> Metamodels.
2. Click **Create Metamodel Repository**.
3. Enter name of repository.
4. Select default and working languages.
5. Click **OK**.



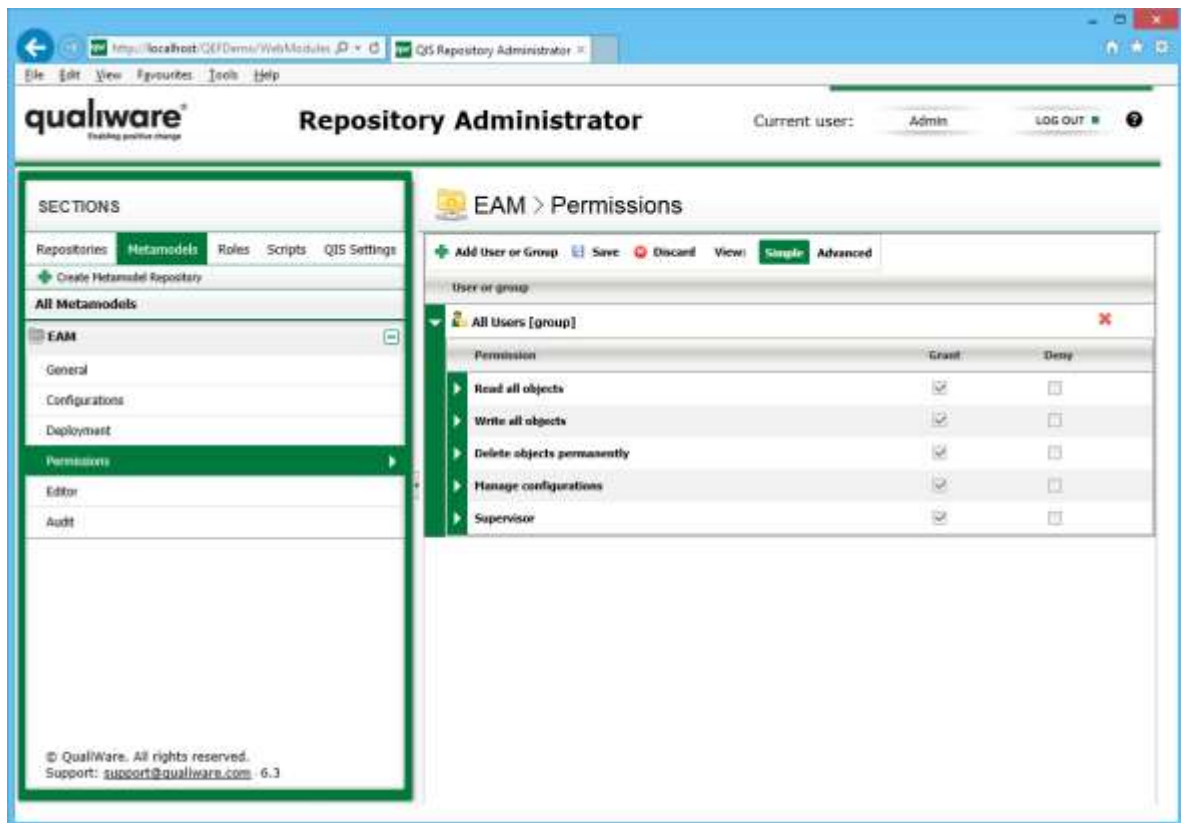
6. Go to RA -> Metamodels -> {Repository} -> General -> Data Storage Settings.
7. Click **Edit**.
8. Click **Generate Settings** or enter settings manually.
9. Click **Create Data Storage**.
10. Click **Save**.



11. Go to RA -> Metamodels -> {Repository} -> Configurations.
12. Enter name of configuration.
13. Click **Create**.



14. Go to RA -> Metamodels -> {Repository} -> Permissions.
15. Select **All Users** group.
16. Select **Supervisor** permission,
17. Click **Save**.

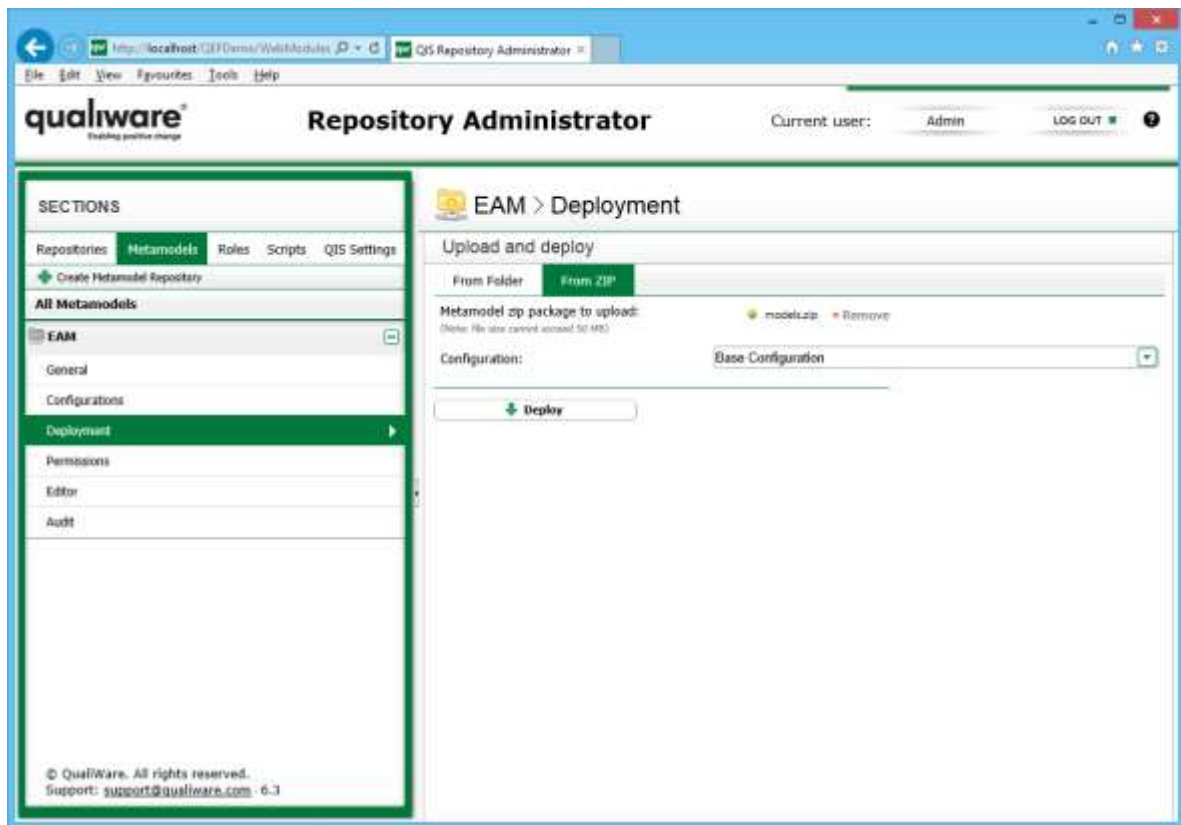


8.2.2.2 Deployment of Metamodel

Deployment can be done either from zip archive or directly from Models folder.

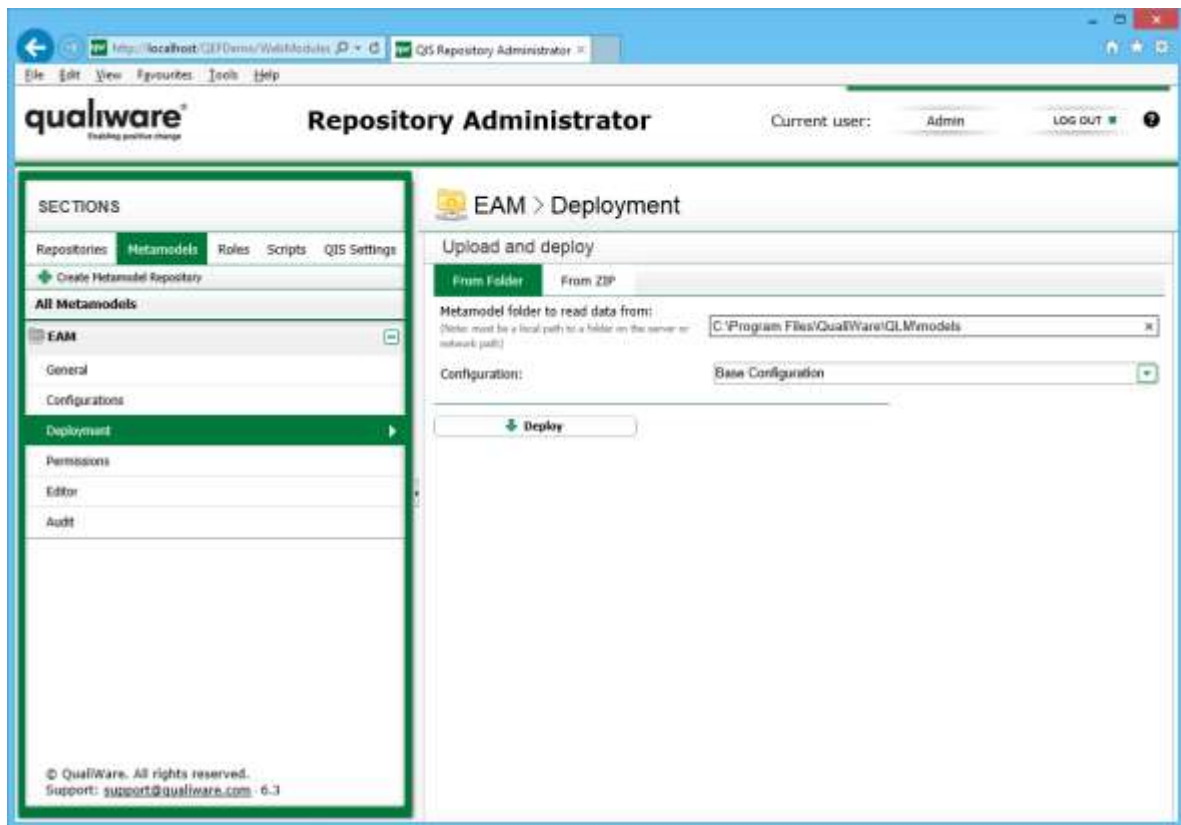
8.2.2.2.1 Deployment of Metamodel from Zip Archive

1. Go to RA -> Metamodels -> {Repository} -> Deployment -> From ZIP.
2. Click **Browse** to select zip archive to deploy.
3. Select configuration.
4. Click **Deploy**.



8.2.2.2.2 Deployment of Metamodel from Models Folder

1. Go to RA -> Metamodels -> {Repository} -> Deployment -> From Folder.
2. Enter full path to Models folder.
3. Select configuration.
4. Click **Deploy**.

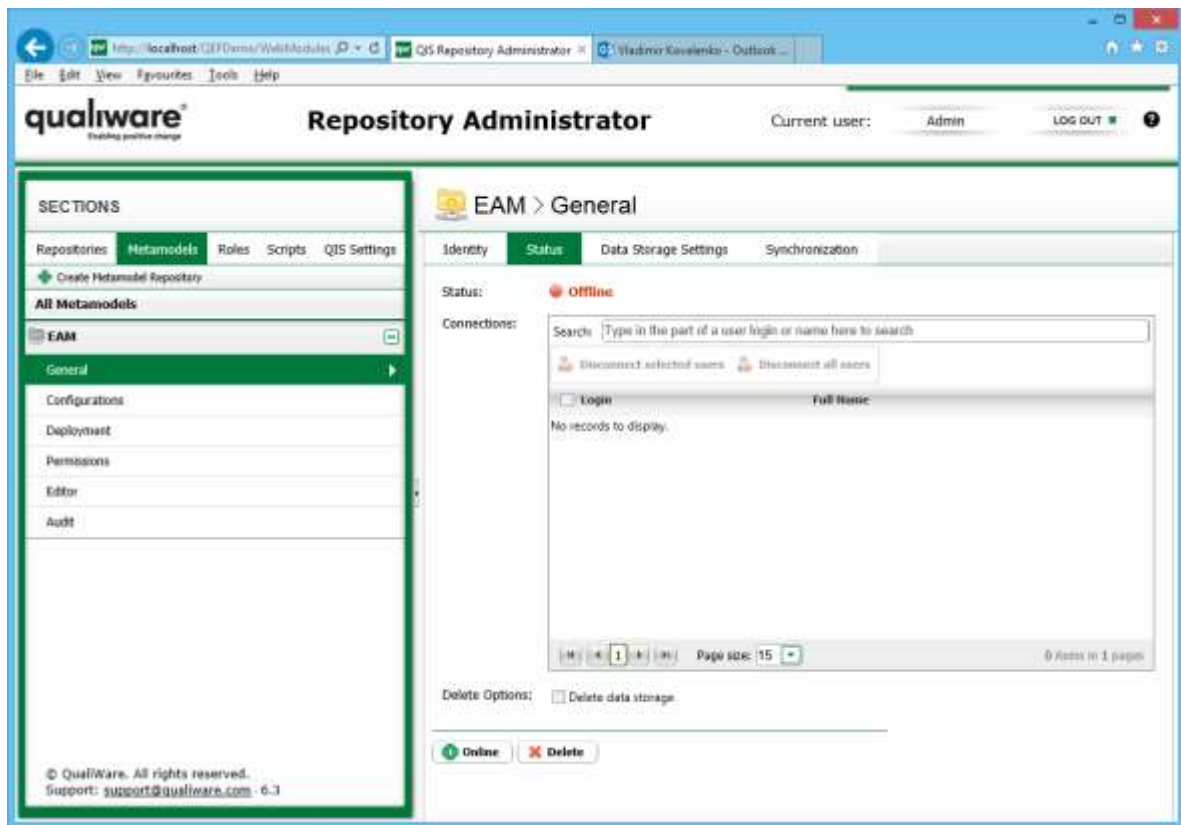


Notes:

- All metamodels found in Models folder will be loaded. With this regard clean it from old and development files (scripts, dialogs, queries etc.).

8.2.2.3 Bring Metamodel Repository Online

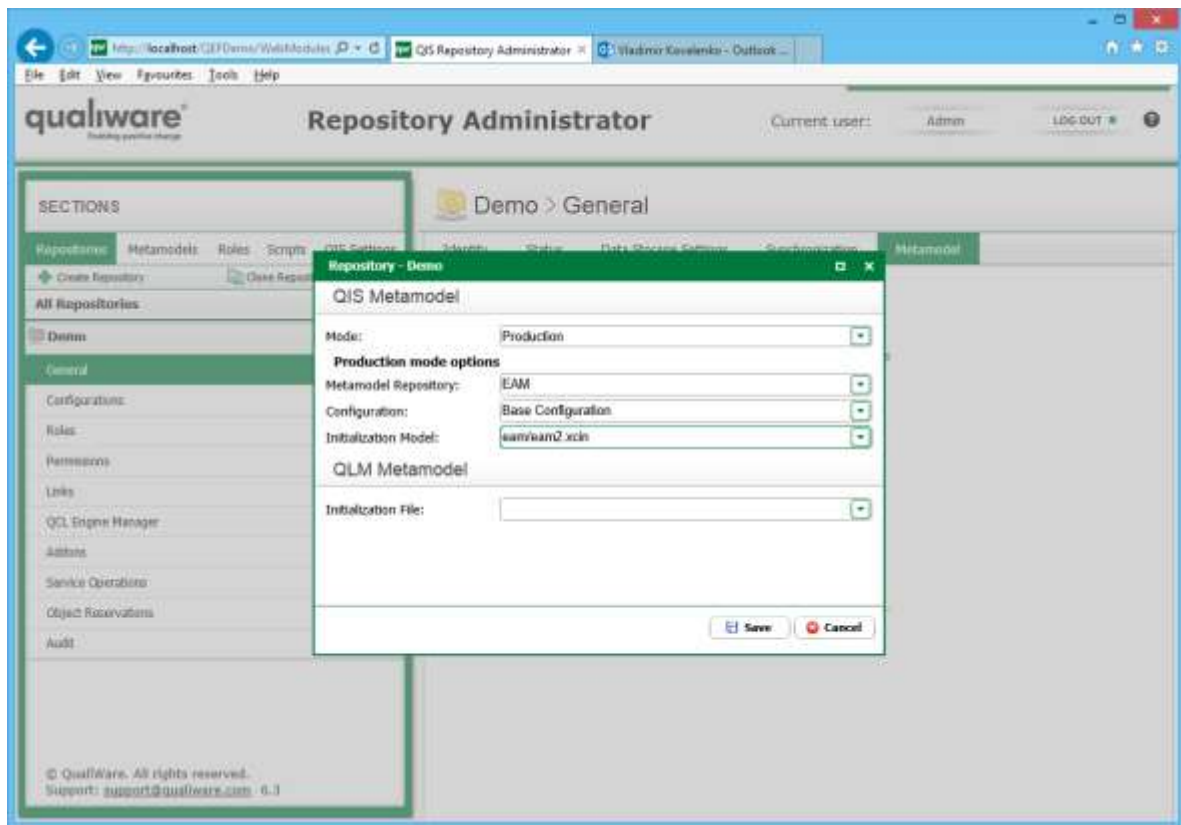
1. Go to RA -> Metamodels -> {Repository} -> General -> Status.
2. Click Online.



8.2.2.4 Assignment of Metamodel

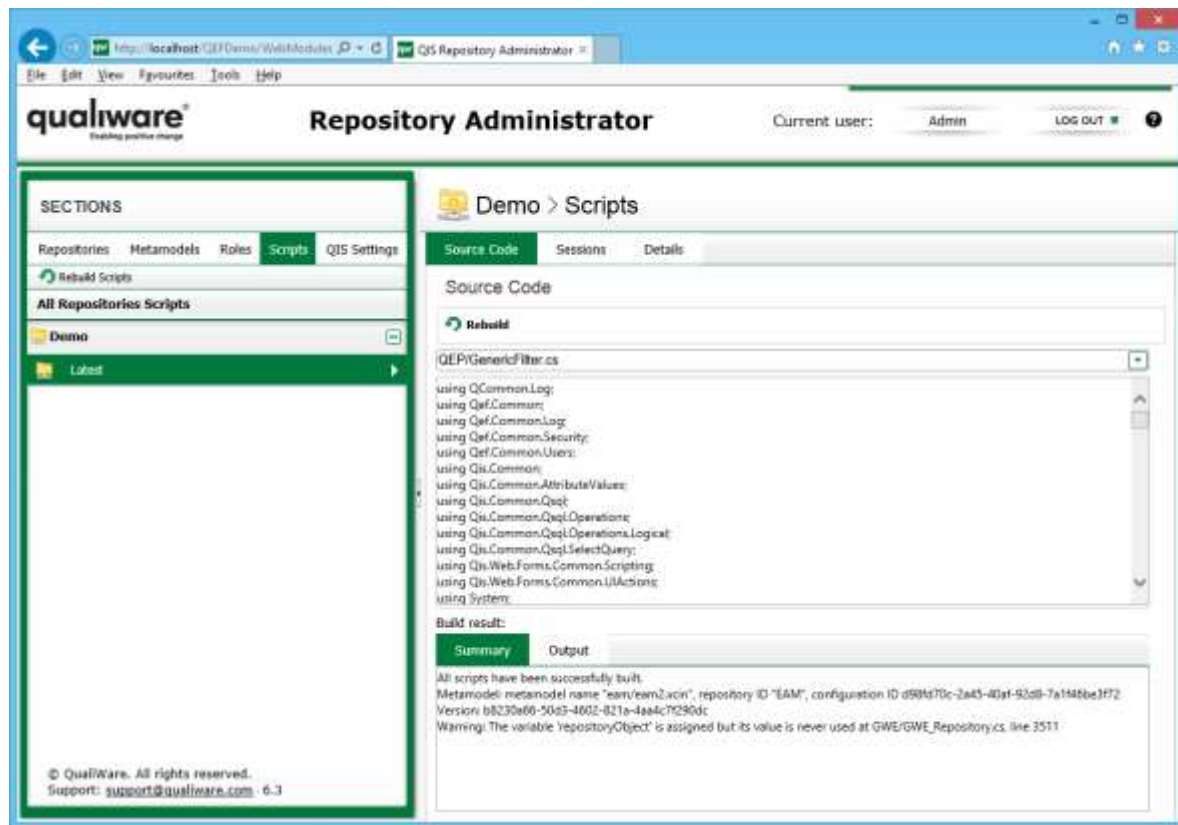
To configure repository to use metamodel in production mode, follow the steps:

1. Go to RA -> Repositories -> {Repository} -> General -> Metamodel.
2. Click **Edit**.
3. Select **Production** mode.
4. Select metamodel repository.
5. Select metamodel configuration.
6. Select .xcin file.
7. Click **Save**.



8.2.3 C# Scripts

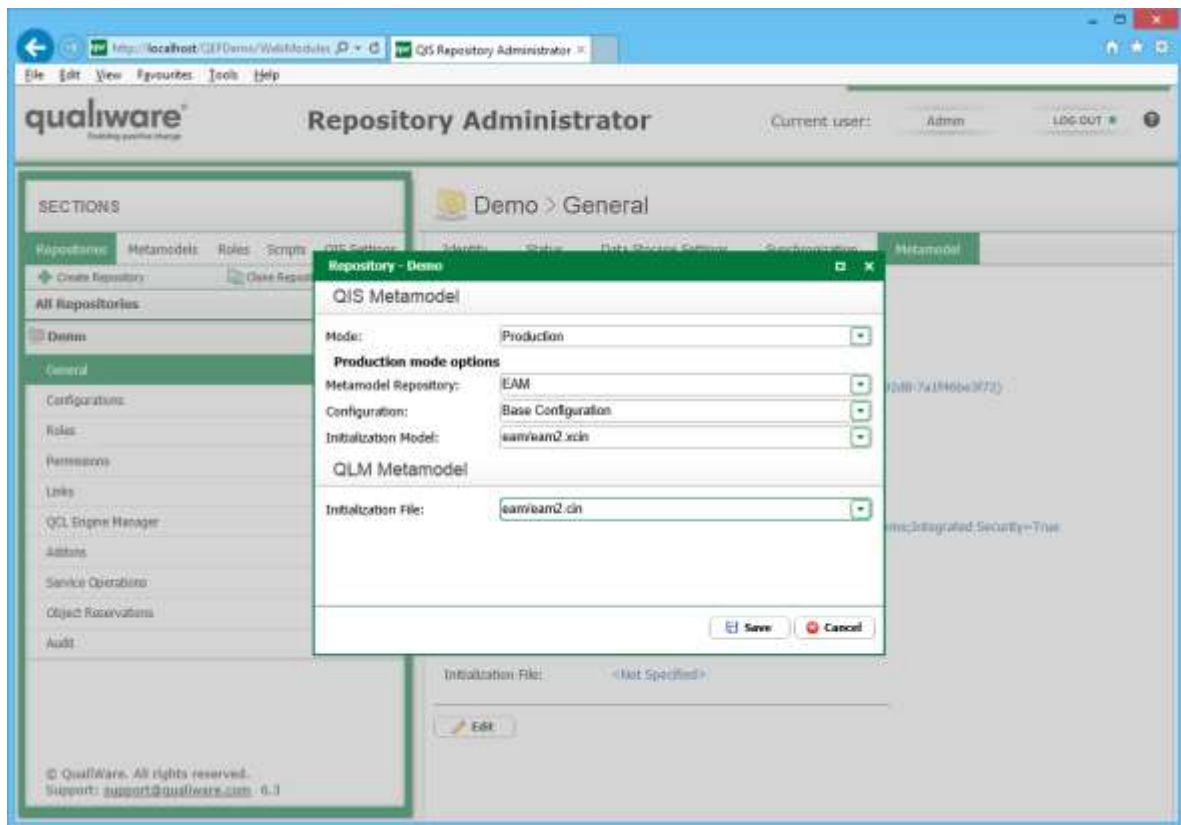
C# Scripts are a part of metamodel. Each repository has its own copy of scripts. Repository scripts can be found at the following location: **RA -> Scripts**. Provided scripts need to be rebuilt, click **Rebuild Scripts**.



8.3 QLM Metamodel

To configure repository to use metamodel in production mode, follow the steps:

1. Go to RA -> Repositories -> [Repository] -> General -> Metamodel.
2. Click **Edit**.
3. Select or enter initialization file.
4. Click **Save**.



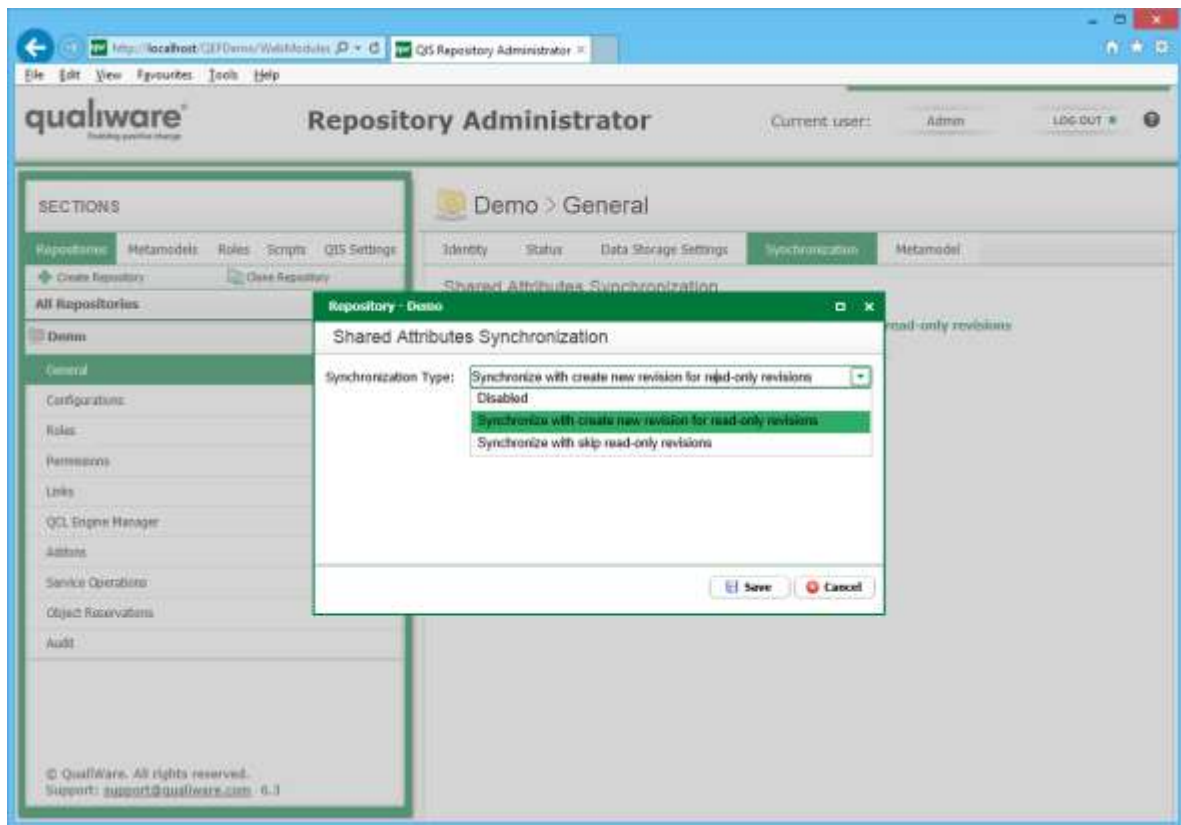
9 Synchronization of Shared Attributes

Object attributes can be marked as shared by metamodel. By default all attributes are not marked as shared. Shared attributes can be synchronized between all language variants of a given object. There are different types of synchronization of shared attributes and they are the following:

- **Synchronize with skip read-only revisions.** Shared attributes are synchronized between read-write language variants only. Read-only language variants are skipped.
- **Synchronize with create new revision for read-only revisions.** Shared attributes are synchronized between read-write and read-only language variants. For read-only language variants new revisions are created.
- **Disabled.** Synchronization of shared attributes is turned off.

To change type of synchronization of shared attributes, follow the steps:

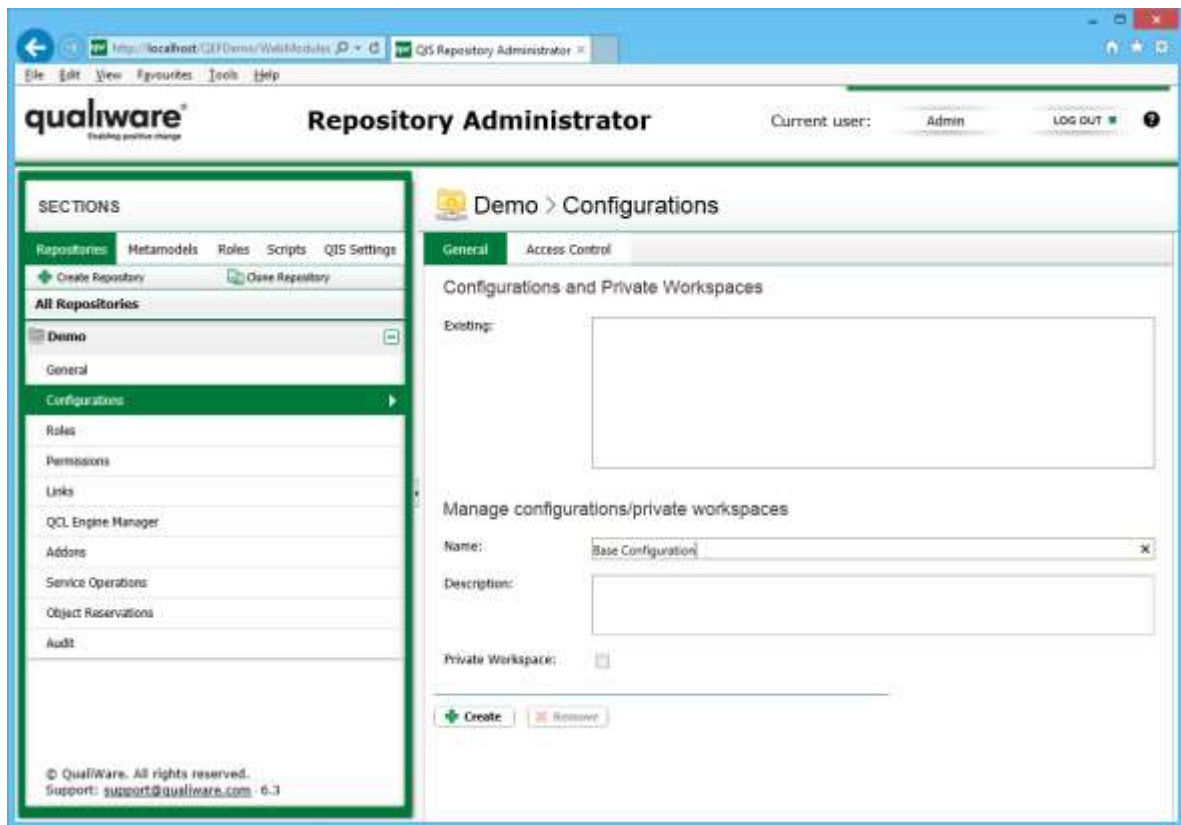
1. Go to RA -> Repositories -> {Repository} -> General -> Synchronization.
2. Click **Edit**.
3. Select synchronization type.
4. Click **Save**.



10 Creation of Configurations

Newly created repository does not contain configurations. Base configuration is created automatically when first connection to repository is made. Or it can be created explicitly from RA. To create configuration, follow the steps:

1. Go to RA -> Repositories -> {Repository} -> Configurations.
2. Enter name and description of configuration.
3. Select if new configuration is a normal configuration or private workspace.
Note: Private workspace can only be created, provided any configuration exists and selected.
4. Click **Create**.



Notes:

- If dump is planned to be loaded in a newly created repository, all existing configurations will be deleted.

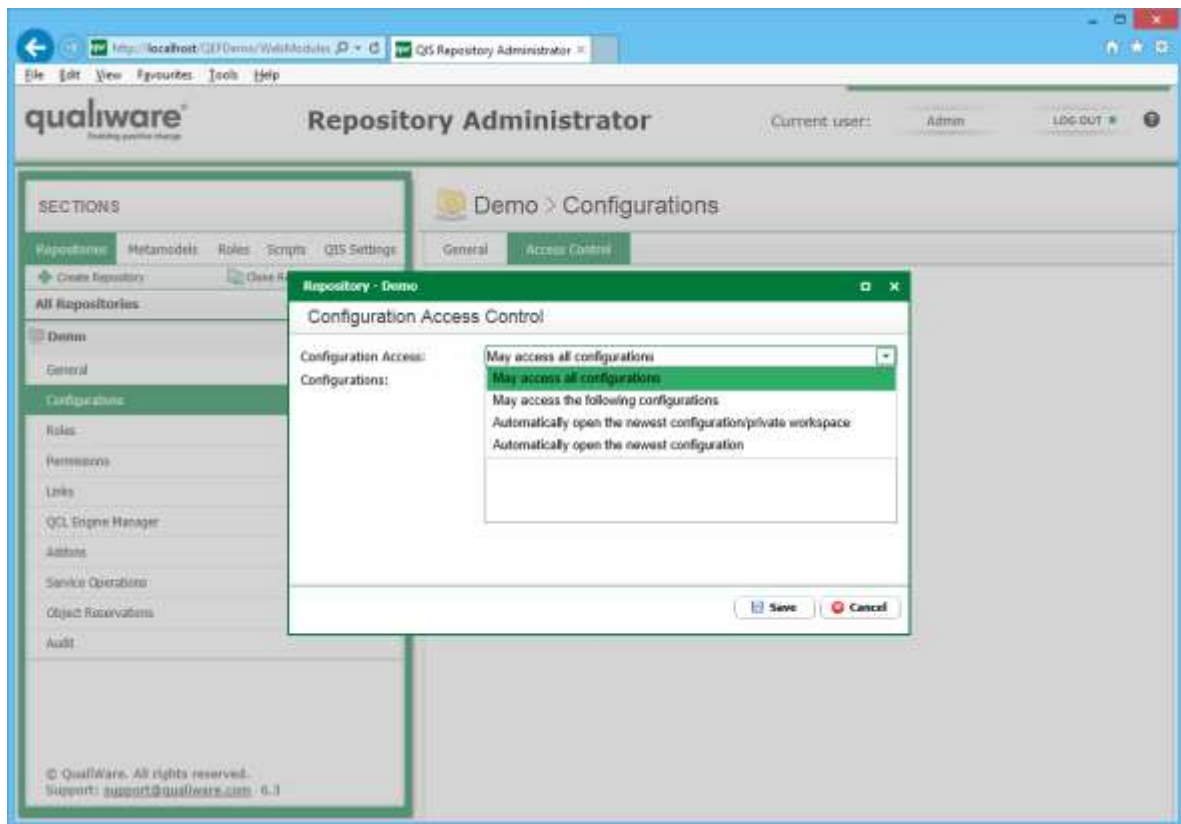
11 Access Control

By default all users have access to all configurations in repository. Access can be refined by access control settings. Access control settings can be defined in order of priority on the following levels: repository, role and remote repository link. There are the following options for access control settings:

- **Use settings for repository** (only available on role level). Inherit settings from repository level.
- **Use settings for role** (only available on remote repository link). Inherit settings from role level.
- **May access all configurations.** User can access all existing configurations.
- **May access the following configurations.** User can access only selected configurations.
- **Automatically open the newest configuration/private workspace.** User is automatically connected to the newest configuration or private workspace.
- **Automatically open the newest configuration.** User is automatically connected to the newest configuration.

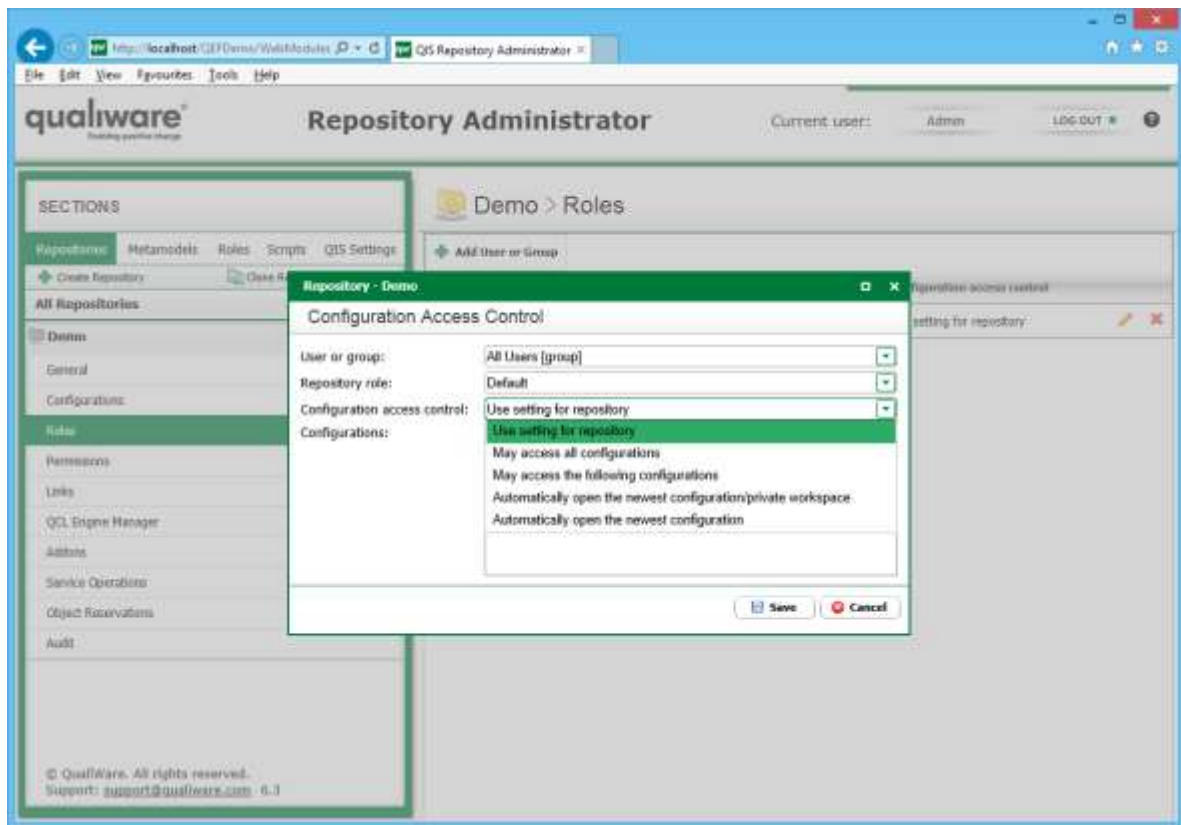
To change access role settings on repository level, follow the steps:

1. Go to **RA -> Repositories -> {Repository} -> Configurations -> Access Control**.
2. Click **Edit**.
3. Select one of configuration access control settings.
4. If **May access the following configurations** setting is selected, select configurations that can be accessed.
5. Click **Save**.



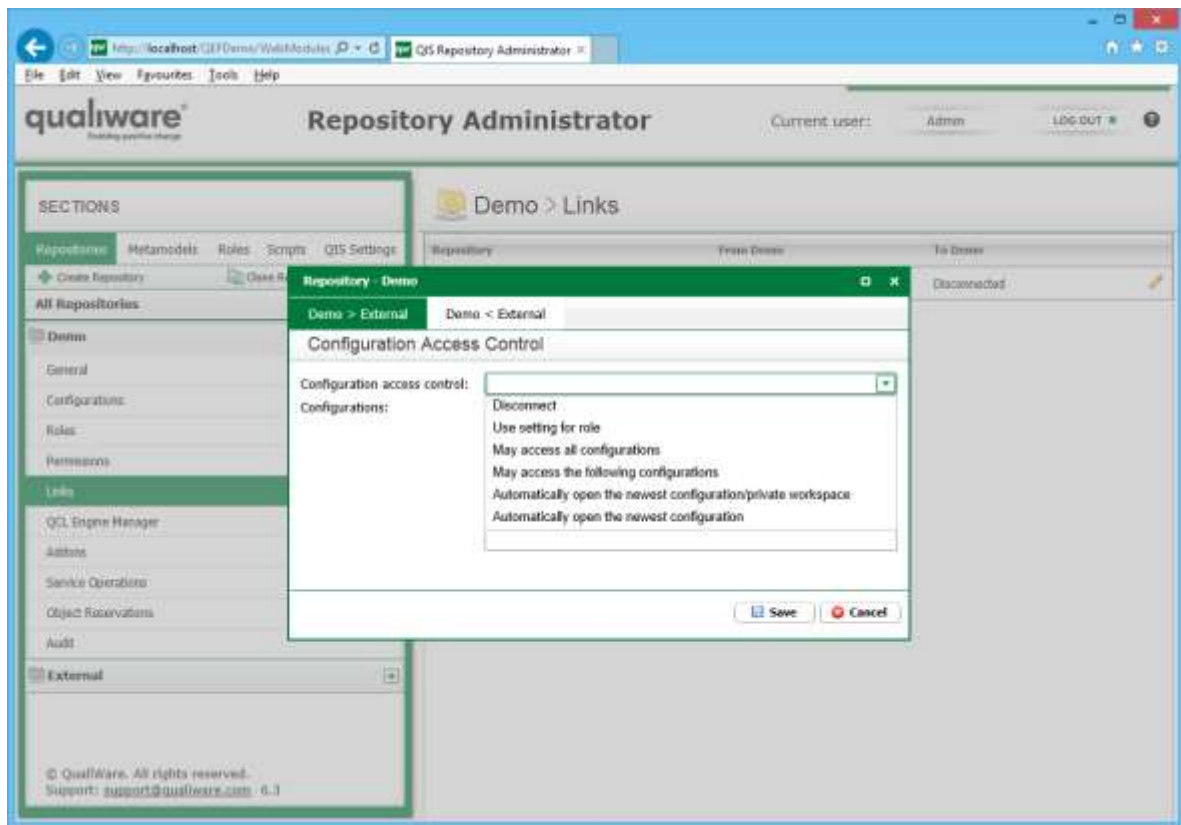
To change access role settings on role level, follow the steps:

6. Go to **RA -> Repositories -> {Repository} -> Roles**.
7. Click **Edit** in front of the role to configure.
8. Select one of configuration access control settings.
9. If **May access the following configurations** setting is selected, select configurations that can be accessed.
10. Click **Save**.



To change access role settings on remote repository link level, follow the steps:

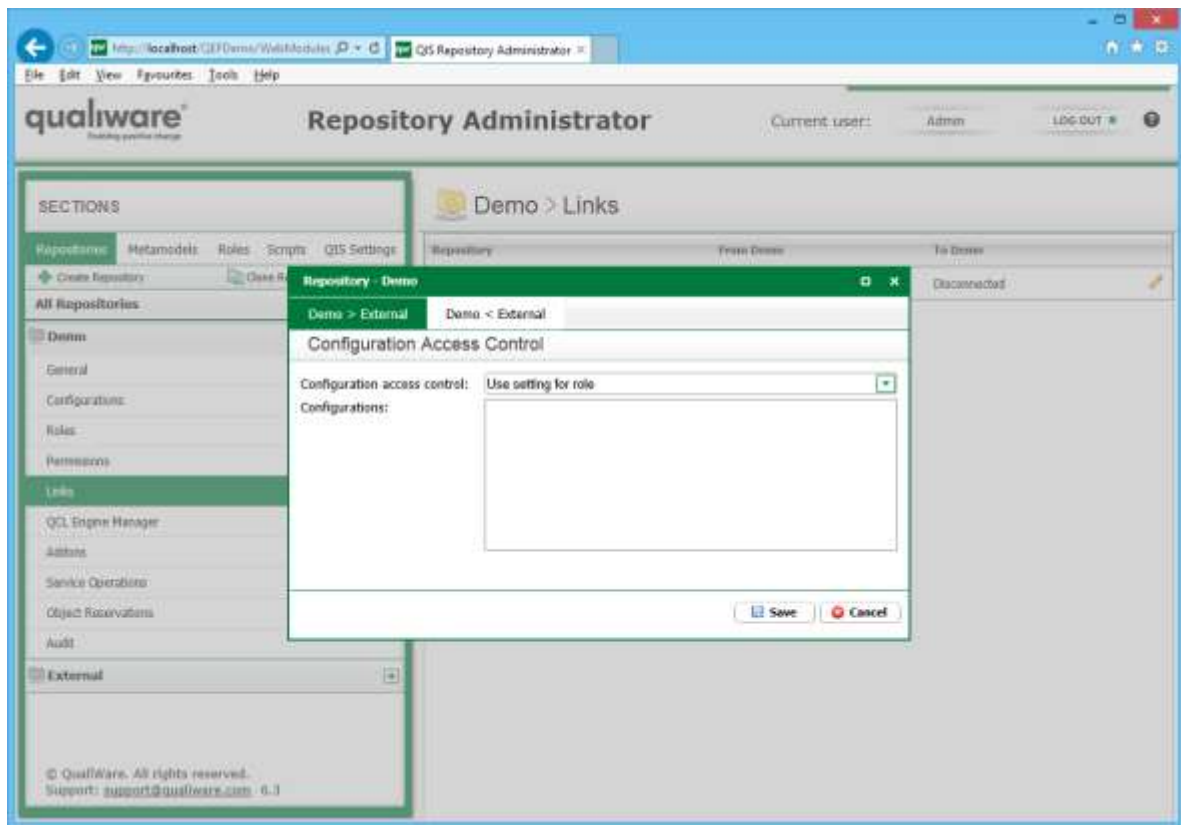
11. Go to **RA -> Repositories -> {Repository} -> Links**.
12. Click **Edit** in front of the remote repository link to configure.
13. Select one of configuration access control settings.
14. If **May access the following configurations** setting is selected, select configurations that can be accessed.
15. Click **Save**.



12 Remote Repositories

By default objects in one repository can only have links to other objects in the same repository. Links to objects in other repositories must be allowed explicitly. To do so, repositories must be linked and connection established. To link repositories, follow the steps:

1. Go to RA -> Repositories -> {Repository} -> Links.
2. Click **Edit** in front of the repository to link.
3. Select one of configuration access control settings for forward and backward link.
4. If **May access the following configurations** setting is selected, select configurations that can be accessed.
5. Click **Save**.



Provided repositories are linked, remote connection can be established using Web Forms, QLM or QIS API and links to objects from other repository can be inserted or opened.

13 QCL Engine Manager

QCL Engine Manager is enabled by installing QEM module in Admin Console.

For detailed documentation on installation, updating and configuration of QEF, see the following documents:

One time configurations inside QEF - Modules installation & updating - QCLE.docx

Repositories and metamodels - One time configuration of repositories - Enabling and configure QCLE.docx

14 Addons

Addons are extensions to the basic functionality provided by QIS. There are the following addons exist: Access Log, Governance Workflow Engine, Social Behavior Warehouse, QChat.

For detailed documentation on configuration of each of the addons, see the following documents:

Repositories and metamodels - One time configuration of repositories - Access Log.docx

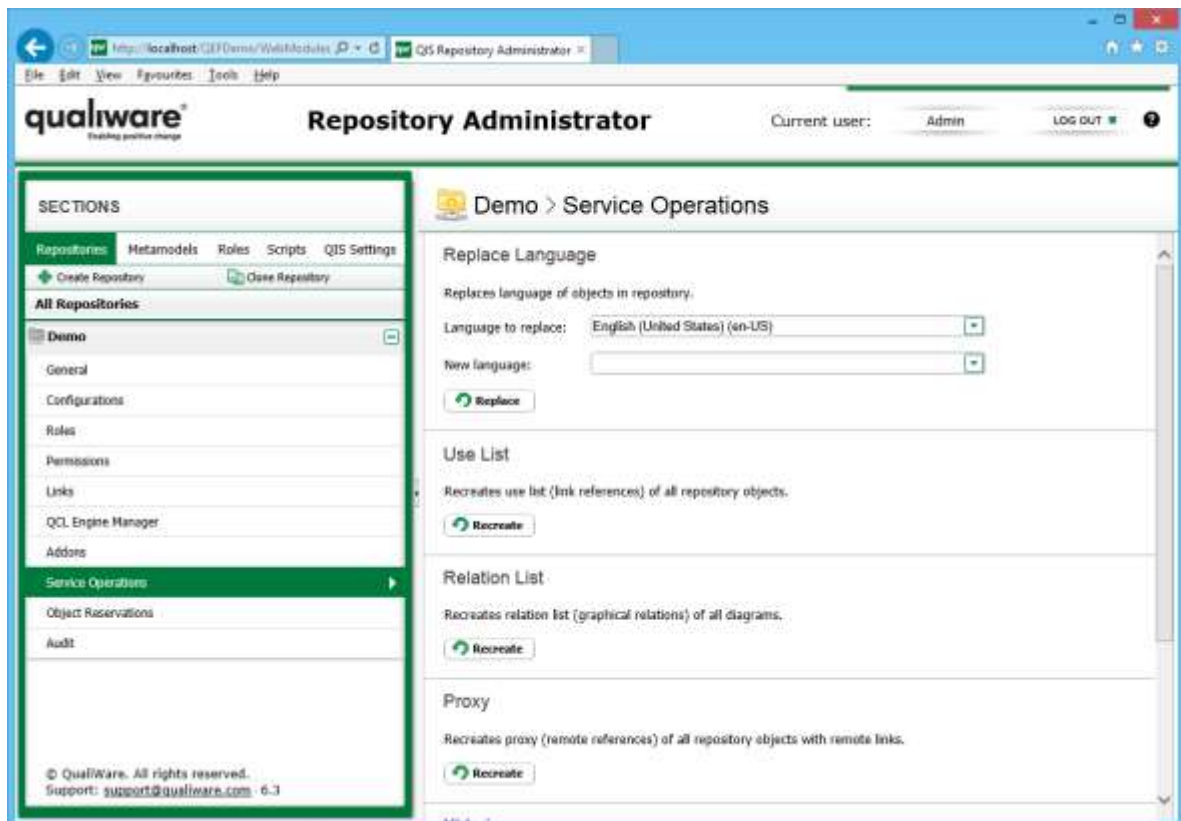
Repositories and metamodels - One time configuration of repositories - Enabling and configure GWE.docx

Repositories and metamodels - One time configuration of repositories - Enabling and configure SBW.docx

15 Service Operations

Service operations are operations designed to fix migration issues and maintenance issues not directly supported by the application. Service operations are available at the following location: **RA -> Repositories -> [Repository] -> Service Operations**. Below is a list of service operations with descriptions:

1. **Replace Language.** It is used to replace language of existing objects. Because change of default language is not supported, it can be worked around with replacing languages.
2. **Use List and Relation List.** It recreates object and graphical links respectively.
3. **Proxy.** Recreates proxy objects in selected repository. Proxy objects are created in local repository when a link to object in remote repository is created.
4. **Xhtml.** Rebuilds index used for full text search in XHTML text.
5. **Private Workspace.** Fixes issue with inheritance of objects from base configuration to private workspace.



16 Object Reservations

Objects in repository are protected from concurrent editing by means of locking and reservation. Object locks are bound to session and are stored in memory on the server. When session terminates, all locks bound to session are released. Object reservations are not bound to session, but user, and are stored in database. When session terminates, reservations are not released and exist until they are explicitly released by user. There are situations when it is needed to forcibly release user locks. To do so, follow the steps:

1. Go to RA -> Repositories -> {Repository} -> Object Reservations.
2. Select user whose reservations to release.
3. Click **Remove reservations for selected users**.

